



# OpenCore

Reference Manual (0.5.~~7~~.8)

[2020.04.09]

- **Overwrite** — Overwrite existing gEfiSmbiosTableGuid and gEfiSmbiosTable3Guid data if it fits new size. Abort with unspecified state otherwise.
- **Custom** — Write ~~first SMBIOS table~~ (SMBIOS tables (~~gEfiSmbiosTableGuid~~gEfiSmbios(3)TableGuid) to ~~gOeCustomSmbiosTableGuid~~gOeCustomSmbios(3)TableGuid to workaround firmwares overwriting SMBIOS contents at ExitBootServices. Otherwise equivalent to **Create**. Requires patching AppleSmbios.kext and AppleACPIPlatform.kext to read from another GUID: "EB9D2D31" - "EB9D2D35" (in ASCII), done automatically by CustomSMBIOSGuid quirk.

*Note: A side effect of using Custom approach is making SMBIOS updates exclusive to macOS, avoiding a collision with existing Windows activation and custom OEM software but potentially breaking Apple-specific tools.*

#### 6. Generic

**Type:** plist dictionary

**Optional:** When Automatic is false

**Description:** Update all fields. This section is read only when Automatic is active.

#### 7. DataHub

**Type:** plist dictionary

**Optional:** When Automatic is true

**Description:** Update Data Hub fields. This section is read only when Automatic is not active.

#### 8. PlatformNVRAM

**Type:** plist dictionary

**Optional:** When Automatic is true

**Description:** Update platform NVRAM fields. This section is read only when Automatic is not active.

#### 9. SMBIOS

**Type:** plist dictionary

**Optional:** When Automatic is true

**Description:** Update SMBIOS fields. This section is read only when Automatic is not active.

## 10.2 Generic Properties

#### 1. SpoofVendor

**Type:** plist boolean

**Failsafe:** false

**Description:** Sets SMBIOS vendor fields to Acidanthera.

It is dangerous to use Apple in SMBIOS vendor fields for reasons given in **SystemManufacturer** description. However, certain firmwares may not provide valid values otherwise, which could break some software.

#### 2. AdviseWindows

**Type:** plist boolean

**Failsafe:** false

**Description:** Forces Windows support in **FirmwareFeatures**.

Added bits to **FirmwareFeatures**:

- FW\_FEATURE\_SUPPORTS\_CSM\_LEGACY\_MODE (0x1) - Without this bit it is not possible to reboot to Windows installed on a drive with EFI partition being not the first partition on the disk.
- FW\_FEATURE\_SUPPORTS\_UEFI\_WINDOWS\_BOOT (0x20000000) - Without this bit it is not possible to reboot to Windows installed on a drive with EFI partition being the first partition on the disk.

#### 3. SystemProductName

**Type:** plist string

**Failsafe:** MacPro6,1

**Description:** Refer to SMBIOS SystemProductName.

#### 4. SystemSerialNumber

**Type:** plist string

**Failsafe:** OPENCORE\_SN1

**Description:** Refer to SMBIOS SystemSerialNumber.

## 12 Troubleshooting

### 12.1 Windows support

#### Can I install Windows?

While no official Windows support is provided, 64-bit UEFI Windows installations (Windows 8 and above) prepared with Boot Camp are supposed to work. Third-party UEFI installations as well as systems partially supporting UEFI boot, like Windows 7, might work with some extra precautions. Things to keep in mind:

- MBR (Master Boot Record) installations are legacy and will not be supported.
- To install Windows, macOS, and OpenCore on the same drive you can specify Windows bootloader path (`\EFI\Microsoft\Boot\bootmgfw.efi`) in `BlessOverride` section.
- All the modifications applied (to ACPI, NVRAM, SMBIOS, etc.) are supposed to be operating system agnostic, i.e. apply equally regardless of the OS booted. This enables Boot Camp software experience on Windows.
- macOS requires the first partition to be EFI System Partition, and does not support the default Windows layout. While OpenCore does have a workaround for this, it is highly recommend not to rely on it and install properly.
- Windows may need to be reactivated. To avoid it consider setting `SystemUUID` to the original firmware UUID. Be warned, on old firmwares it may be invalid, i.e. not random. In case you still have issues, consider using HWID or KMS38 license ~~or making the use Custom UpdateSMBIOSMode~~. Other nuances of Windows activation are out of the scope of this document and can be found online.

#### What additional software do I need?

To enable operating system switching and install relevant drivers in the majority of cases you will need Windows support software from Boot Camp. For simplicity of the download process or when configuring an already installed Windows version a third-party utility, Brigadier, can be used successfully. Note, that you may have to download and install 7-Zip prior to using Brigadier.

Remember to always use the latest version of Windows support software from Boot Camp, as versions prior to 6.1 do not support APFS, and thus will not function correctly. To download newest software pass most recent Mac model to Brigadier, for example `./brigadier.exe -m iMac19,1`. To install Boot Camp on an unsupported Mac model afterwards run PowerShell as Administrator and enter `msiexec /i BootCamp.msi`. In case you already have a previous version of Boot Camp installed you will have to remove it first by running `msiexec /x BootCamp.msi` command. `BootCamp.msi` file is located in `BootCamp/Drivers/Apple` directory and can be reached through Windows Explorer.

While Windows support software from Boot Camp solves most of compatibility problems, sometimes you may have to address some of them manually:

- To invert mouse wheel scroll direction `FlipFlopWheel` must be set to 1 as explained on SuperUser.
- `RealTimeIsUniversal` must be set to 1 to avoid time desync between Windows and macOS as explained on SuperUser (this one is usually not needed).
- To access Apple filesystems like HFS and APFS separate software may need to be installed. Some of the known tools are: Apple HFS+ driver (hack for Windows 10), HFSExplorer, MacDrive, Paragon APFS, Paragon HFS+, TransMac, etc. Remember to never ever attempt to modify Apple file systems from Windows as this often leads to irrecoverable data loss.

#### Why do I see Basic data partition in Boot Camp Startup Disk control panel?

Boot Camp control panel uses GPT partition table to obtain each boot option name. After installing Windows separately you will have to relabel the partition manually. This can be done with many tools including open-source `gdisk` utility. Reference example:

---

```
PS C:\gdisk> .\gdisk64.exe \.\physicaldrive0
GPT fdisk (gdisk) version 1.0.4
```

```
Command (? for help): p
Disk \.\physicaldrive0: 419430400 sectors, 200.0 GiB
Sector size (logical): 512 bytes
Disk identifier (GUID): DEC57EB1-B3B5-49B2-95F5-3B8C4D3E4E12
```